基于改进粒子群算法的四旋翼自抗扰控制器优化设计

胡丹丹，张宇辰

(中国民航大学 机器人研究所，天津 300300)

摘　要：针对四旋翼飞行器自抗扰控制器参数较多，人工整定困难且难以得到最优控制效果的问题，提出一种基于改进粒子群算法的四旋翼自抗扰控制器优化方法。在设计了四旋翼飞行器的自抗扰控制器之后，将自抗扰控制器的参数作为粒子群中的粒子进行迭代寻优，同时在传统的粒子群算法基础上，参考遗传算法，对适应值不好的粒子进行交叉保优，以提高粒子的多样性，加快寻优速度。仿真结果表明，对比人工整定参数的控制器，优化后的控制器超调更小，调节时间更快。该方法能够解决四旋翼飞行器自抗扰控制器人工参数整定困难的问题，且优化后的控制器具有更好的控制效果。

关键词：四旋翼飞行器；自抗扰控制；粒子群算法；参数优化；姿态控制；

中图分类号：TP242　　doi: 10.3969/j.issn.1001-3695.2018.01.0026

ADRC controller optimization design based on improved PSO algorithm for quad-rotor

Hu Dandan, Zhang Yuchen

(*Robotics Institute*, *Civil Aviation University of China*, *Tianjin* 300300, *China*)

Abstract: In order to solve the problem that the parameters of quad-rotor aircraft ADRC controller are difficult to set, and it is hard to get the optimal control effect due to manual parameter tuning, this paper proposed a design method of adaptive disturbance rejection controller based on improved particle swarm optimization for quad-rotor. After designed the ADRC controller of quad-rotor, this method iteratively optimized the parameters of the controller as particles in the particle swarm. At the same time, based on the traditional particle swarm optimization and referring to genetic algorithm, it cross-guaranteed the particles with poor fitness value to Improve the diversity of particles, and to speed up the search speed. The simulation results show that compared with the controller with manual tuning parameters, the optimized controller has less overshoot and faster adjusting time. This method can solve the difficulty of manual parameter setting of quad-rotor aircraft ADRC controller, and the optimized controller has better control performance.

Key words: quad-rotor aircraft; active disturbance rejection controller(ADRC); particle swarm optimization(PSO); parameter optimization; attitude control

0　引言

四旋翼飞行器的控制系统是一个四输入、六输出的欠驱动非线性系统，具有强耦合性特点，难以建立精确模型且易受到外界干扰，因此，其优化设计成为近年来国内外学者研究的热点和难点。

Gonzalez-vazquez等人[1]于2010年基于经典PID控制方法，采用非线性PID控制器实现了四旋翼飞行器的姿态控制和位置控制。Alothman等人[2]于2015年提出了一种基于LQR的四旋翼飞行器载荷运输系统的设计方法，仿真结果表明控制效果优于传统PID控制器。Yang等人[3]提出了一种基于动态逆的四旋翼飞行器非线性控制方法。为了补偿该方法所产生的误差，设计了基于神经网络的自适应动态反转控制器。Raffo等[4]在2015年提出了一种用于四旋翼飞行器路径跟踪的非线性鲁棒控制策略，仿真结果表明该方法具有良好的鲁棒性。

诸如LQR、鲁棒控制、反步控制等[5,6]控制策略虽然有比较好的效果，但由于其控制算法或需要基于准确的数学模型，或计算量很大，数据的实时处理对处理器性能要求高，实现起来有一定的困难。目前各类四旋翼飞行器依然是使用传统串级PID作为核心控制算法，PID算法对于控制精度不高的飞行任务可以基本满足控制需求，但是其抗干扰能力差，且无法满足高精度的控制需求。

为了提高飞行器的抗干扰性，学者们开始使用自抗扰技术来设计旋翼控制器[7~9]，相比于传统串级PID控制器拥有更好

收稿日期：2018-01-17；**修回日期：**2018-02-28

作者简介：胡丹丹（1979-），女，吉林柳河人，副教授，硕士，主要研究方向为机器人感知与认知（348161344@qq.com）；张宇辰（1990-），男，天津人，硕士研究生，主要研究方向为多旋翼飞行器的自主控制.

的控制效果。但是自抗扰控制器参数较多，整定困难。为了得到最佳控制效果，且解决人工参数整定困难的问题，本文首先基于自抗扰技术设计了四旋翼飞行器控制器，对于自抗扰控制器的参数使用粒子群算法进行寻优，同时对于粒子群算法易陷入局部最优的问题加入遗传算法中的交叉保优思想进行改进。

1　四旋翼飞行器数学模型建立

选取 X 型四旋翼机型建立数学模型。定义两个坐标系：惯性坐标系和机体坐标系，如图 1 所示。

惯性坐标系 E（OXYZ）相对于地球表面不动，取“东北天”（ENU）建立坐标系。

机体坐标系 B（oxyz）与飞行器固联，原点为飞行器的重心、质心，ox 轴为 1 号与 3 号电机夹角的角平分线方向，oy 轴为 2 号与 3 号电机夹角的角平分线方向，ox 轴垂直于 oxy 平面指向机体上方，该坐标系符合右手定则。

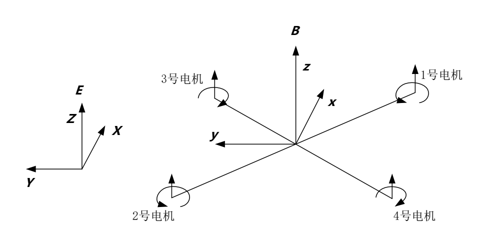


图 1　惯性坐标系与机体坐标系示意图

考虑到四旋翼飞行器的气动环境复杂，且易受到未知的外界干扰，为了方便模型的建立，对飞行器不失一般性地作出如下几点假设：a)视四旋翼飞行器为均匀对称刚体，飞行过程中质量保持不变；b)机体坐标系的原点与飞行器几何中心及质心始终重合；c)由于飞行器不会飞的很高很远，所以视地球表面为平面，飞行器所受重力和阻力不受飞行器飞行高度变化的影响，始终保持不变；d)不计算地球自转与公转对飞行器的影响。

飞行器的各姿态角定义如图 2 所示。

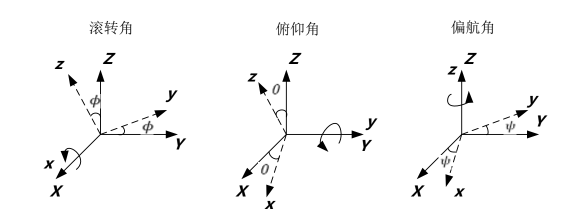


图 2　飞行器姿态角示意图

基于以上设定，四旋翼飞行器动力学模型可表示为如下形式：

$$\begin{cases} \ddot{x}=\left(\cos\phi\sin\theta\cos\psi+\sin\phi\sin\psi\right)\dfrac{U_1}{m}+\Delta_x \\ \ddot{y}=\left(\cos\phi\sin\theta\sin\psi+\sin\phi\cos\psi\right)\dfrac{U_1}{m}+\Delta_y \\ \ddot{z}=\left(\cos\phi\cos\theta\right)\dfrac{U_1}{m}-g+\Delta_z \\ \ddot{\phi}=\dot{\theta}\dot{\psi}\left(\dfrac{I_y-I_z}{I_x}\right)+\dfrac{l}{\sqrt{2}I_x}U_2+\Delta_\phi \\ \ddot{\theta}=\dot{\phi}\dot{\psi}\left(\dfrac{I_z-I_x}{I_y}\right)+\dfrac{l}{\sqrt{2}I_y}U_3+\Delta_\theta \\ \ddot{\psi}=\dot{\phi}\dot{\theta}\left(\dfrac{I_x-I_y}{I_z}\right)+\dfrac{U_4}{I_z}+\Delta_\psi \end{cases}$$

定义 $U_1\sim U_4$ 为各个通道的控制量，表述如下：

$$\begin{cases} U_1=F_1+F_2+F_3+F_4=k_1\left(\Omega_1^2+\Omega_2^2+\Omega_3^2+\Omega_4^2\right) \\ U_2=-F_1+F_2+F_3-F_4=k_1\left(-\Omega_1^2+\Omega_2^2+\Omega_3^2-\Omega_4^2\right) \\ U_3=-F_1+F_2-F_3+F_4=k_1\left(-\Omega_1^2+\Omega_2^2-\Omega_3^2+\Omega_4^2\right) \\ U_4=F_1+F_2-F_3-F_4=k_2\left(\Omega_1^2+\Omega_2^2-\Omega_3^2-\Omega_4^2\right) \end{cases}$$

其中：ϕ,θ,ψ 分别为滚转角、俯仰角和偏航角，m 为飞行器的质量，g 为重力加速度，I_x,I_y,I_z 分别为三个轴上的转动惯量，l 为旋翼中心到机体坐标系原点的距离，$F_1\sim F_4$ 为对应旋翼升力，$\Omega_1\sim\Omega_4$ 为对应电机的转速，k_1 为旋翼的升力系数，k_2 为旋翼的阻力系数。$\Delta_x,\Delta_y,\Delta_z,\Delta_\phi,\Delta_\theta,\Delta_\psi$ 分别表示各个通道未建模和干扰部分。

2　四旋翼飞行器自抗扰控制器设计

2.1　自抗扰控制器结构设计

自抗扰控制技术（active disturbance rejection controller，ADRC）是韩京清教授于上世纪 80 年代末期提出的一种新的控制理论[10]，其特点是不依赖于控制对象的模型，可以将系统建模不准确产生的“内扰”和系统外部扰动结合起来进行估计补偿。其结构包括跟踪微分器（TD）、扩张状态观测器（ESO）和非线性误差反馈控制律（NLSEF）三个部分。二阶自抗扰控制器结构如图 3 所示。

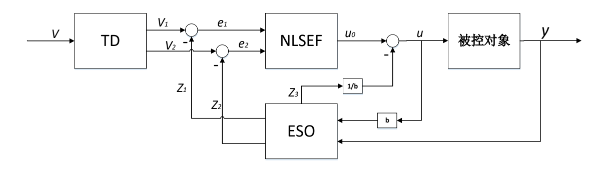


图 3　二阶自抗扰控制器结构图

四旋翼飞行器高度与姿态控制结构图如图 4 所示。

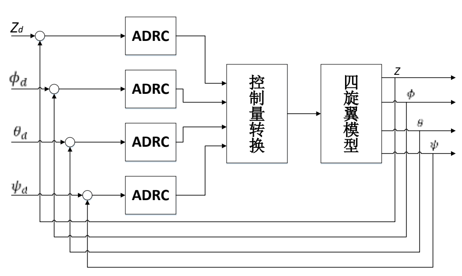


图 4　四旋翼飞行器高度与姿态控制结构图

姿态控制是飞行器可以稳定飞行的基础，由飞行器动力学模型可知，四旋翼的姿态角通道耦合在一起构成了飞行器的“内扰”，这可以通过设计扩张状态观测器来进行估计补偿，从而实现解耦控制，使得各个通道变为“积分串联器”型的线性系统[11]。姿态系统方程如下：

$$\begin{cases} \ddot{\phi}=\dot{\theta}\dot{\psi}\left(\dfrac{I_y-I_z}{I_x}\right)+\dfrac{l}{\sqrt{2}I_x}U_2+\Delta_\phi \\ \ddot{\theta}=\dot{\phi}\dot{\psi}\left(\dfrac{I_z-I_x}{I_y}\right)+\dfrac{l}{\sqrt{2}I_y}U_3+\Delta_\theta \\ \ddot{\psi}=\dot{\phi}\dot{\theta}\left(\dfrac{I_x-I_y}{I_z}\right)+\dfrac{U_4}{I_z}+\Delta_\psi \end{cases}$$

将上式整理为自抗扰理论对应的形式：

$$\begin{bmatrix}\ddot{\phi}\\\ddot{\theta}\\\ddot{\psi}\end{bmatrix}=\begin{bmatrix}f_1(\phi,\theta,\psi,\Delta)\\f_2(\phi,\theta,\psi,\Delta)\\f_3(\phi,\theta,\psi,\Delta)\end{bmatrix}+\boldsymbol{B}\begin{bmatrix}U_2\\U_3\\U_4\end{bmatrix}$$

其中：$\boldsymbol{B}=\begin{bmatrix}\frac{l}{\sqrt{2}I_x}&0&0\\0&\frac{l}{\sqrt{2}I_y}&0\\0&0&\frac{1}{I_z}\end{bmatrix}$，$\begin{cases}f_1=\dot{\theta}\dot{\psi}\left(\frac{I_y-I_z}{I_x}\right)+\Delta_\phi\\f_2=\dot{\phi}\dot{\psi}\left(\frac{I_z-I_x}{I_y}\right)+\Delta_\theta\\f_3=\dot{\phi}\dot{\theta}\left(\frac{I_x-I_y}{I_z}\right)+\Delta_\psi\end{cases}$。

2.2 自抗扰控制器算法设计

以四旋翼飞行器的滚转通道为例来描述自抗扰控制器的设计过程。T 为控制器的控制间隔。

1）跟踪微分器（TD）

TD 可以为输入信号安排过渡过程，并提取输入信号的微分，表示如下：

$$\begin{cases}v_1(k+1)=v_1(k)+Tv_2(k)\\v_2(k+1)=v_2(k)+T\,\text{fst}(v_1(k)-v_d,v_2(k),r,h)\end{cases}$$

其中 v_d 是期望值，r 为速度因子，h 为滤波因子，提高 r 值可以减小安排过渡过程的时间，过大的 r 值会使得跟踪信号更加逼近输入信号，但会失去安排过渡过程的意义。滤波因子 h 可以滤掉跟踪信号中的噪声，h 越大滤波效果越好，但是增大 h 会产生输入信号相位损失，一般来讲 h 设定为 3-10 倍的控制周期。

2）扩张状态观测器（ESO）

ESO 对系统的总扰动进行估计，并且实时补偿给控制器，其表示如下：

$$\varepsilon_1=z_1(k)-y(k)$$
$$\begin{cases}z_1(k+1)=z_1(k)+T(z_2(k)-\beta_1\varepsilon_1)\\z_2(k+1)=z_2(k)+T(z_3(k)-\beta_2\,\text{fal}(\varepsilon_1,\alpha_1,\delta)+bu(k))\\z_3(k+1)=z_3(k)-T\beta_3\,\text{fal}(\varepsilon_1,\alpha_2,\delta)\end{cases}$$

其中：α_1,α_2 决定 $\text{fal}(\bullet)$ 的非线性程度，一般的，α_1 取 0.5，α_2 取 0.25。δ 为 $\text{fal}(\bullet)$ 函数在原点附近的线性区间宽度，如果 δ 太大会降低 ESO 对于非线性型号的逼近能力，过小的 δ 会导致高频震荡。β_1,β_2,β_3 为状态误差反馈的反馈增益，影响 ESO 的收敛速度。β_1 越大 z_1 跟踪输入信号的速度越快。同样的，β_2 越大 z_2 跟踪输入信号的速度越快。β_3 关系着观测器对于扰动估计产生的滞后，β_3 越大产生的滞后越小。β_1,β_2,β_3 三个参数之间相互制约，增大 β_1,β_2 能够抑制 β_3 过大产生的震荡。

3）非线性误差反馈控制律（NLSEF）

NLSEF 中反馈误差以非线性的组合计算控制量，表示如下：

$$\begin{cases}e_1=v_1(k)-z_1(k)\\e_2=v_2(k)-z_2(k)\\u_0(k)=k_1\,\text{fal}(e_1,\alpha_1,\delta_0)+k_2\,\text{fal}(e_2,\alpha_2,\delta_0)\\u(k)=u_0(k)-\frac{z_3(k)}{b}\end{cases}$$

NLSEF 中要满足 $0<\alpha_1<1<\alpha_2$ 的条件，一般取 α_1 为 0.75，α_2 为 1.25。δ_0 的调节与 ESO 中的 δ 一致。k_1 越大系统响应越快，k_2 可以抑制系统的超调。b 由系统状态方程决定，对于四旋翼飞行器无法精确建模的特性，可以将 b 作为一个可调参数。算法中 $\text{fal}(\bullet)$ 和 $\text{fst}(\bullet)$ 表示如下：

$$\text{fal}(\varepsilon,\alpha,\delta)=\begin{cases}|\varepsilon|^\alpha\,\text{sign}(\varepsilon),|\varepsilon|>\delta,\\\frac{\varepsilon}{\delta^{1-\alpha}},|\varepsilon|\le\delta,\end{cases}\quad\delta>0$$

$$\text{fst}(x_1,x_2,r,h)=-\begin{cases}\frac{r\alpha}{d},|\alpha|\le d\\r\,\text{sign}(\alpha),|\alpha|>d\end{cases}$$

$$d=rh,d_0=dh,y=x_1+hx_2,\alpha_0=\left(d^2+8r|y|\right)^{1/2}$$

$$\alpha=\begin{cases}x_2+\frac{a_0-d}{2},|y|>d_0\\x_2+\frac{y}{h},|y|\le d_0\end{cases}$$

自抗扰控制器不依赖模型，不需要针对具体被控对象来分析系统的稳定性，只需选取适当参数，就可以实现四旋翼飞行器姿态和高度的稳定控制。

3 基于改进粒子群算法的自抗扰控制器参数优化

粒子群算法（particle swarm optimization，PSO）是计算智能领域的一种群体智能优化算法，由 Kennedy 和 Eberhart 博士在 1995 年提出的[12,13]。该算法已广泛的应用于众多领域的参数寻优问题[14,15]。粒子群算法速度与位置更新公式表示如下：

$$\begin{cases}v_{id}(k+1)=\omega v_{id}(k)+c_1r_1(p_{id}-z_{id}(k))\\\qquad+c_2r_2(p_{gd}-z_{id}(k))\\z_{id}(k+1)=z_{id}(k)+v_{id}(k+1)\end{cases}$$

其中：k 为迭代次数，z_i 为第 i 个粒子，d 为每个粒子的维度，p_i 为粒子迄今搜索到的最后位置，p_g 为整个粒子群迄今搜索到的最优位置，c_1,c_2 为粒子的学习因子，使得粒子具有自我总结和向群体中优秀粒子学习的能力，ω 为粒子惯性权重，起着权衡局部最优和全局最优的作用，r_1,r_2 为 $[0,1]$ 之间的随机数，使得群体具有多样性。

传统的粒子群算法是通过追随个体极值和群体极值来完成寻优的，虽然操作简单，且能够很快的收敛，但是随着迭代次数的增加，在收敛的同时，各粒子越来越相似，可能在局部解周边无法跳出。为了解决粒子群算法在进行寻优计算过程中陷入到局部最优的问题，借鉴遗传算法，在粒子每次迭代寻优的过程中加入选择、交叉操作以对粒子群算法进行改进，使得算法可以遍历到更多的粒子。具体为：每一次迭代后计算粒子群适应值的算术平均值，取适应值大于算术平均值的最大偶数个粒子直接进入下一次迭代，将剩余粒子进行遗传算法中的交叉选择操作，随机选取需要交叉的位数，并对粒子对应位进行交叉，将交叉后生成的子代与父代结合在一起并比较适应值，选择适应值更好的一半进入下一次迭代，生成新一组粒子群。

算法的适应度函数选取 ITAE 误差准则，其值越低，性能越好，基本形式如下：

$$J=\int_0^\infty t|e(t)|dt$$

在自抗扰控制器中有 14 个可调参数，其中 ESO 中的

β_1,β_2,β_3 和 NLSEF 中的 k_1,k_2,b 是控制器最关键的 6 个参数，前者决定扰动的估计补偿能力，后者直接决定控制量的大小，本文将这 6 个参数作为粒子群算法的粒子，设计自抗扰控制器粒子群算法。

改进粒子群算法的流程如下所述：

a)初始化粒子群，随机产生所有粒子的位置和速度，并确定粒子的 p_i 和 p_g ；

b)按照粒子群速度与位置公式更新粒子的速度和位置；

c)将每个粒子代入控制器仿真器并计算对应的 ITAE 值。

d）计算所有粒子 ITAE 值的算术平均值，取偶数个 ITAE 值小于平均值的粒子直接进入下一次迭代。剩余粒子进行交叉选择，生成相同数量的子代，并与父代一起进行排序，取 ITAE 值更小的一半进入下一次迭代。

e)将每一个粒子的适应值与该粒子所经历过的最优位置 p_i 的适应值进行比较，如果更好，则将该粒子其作为新的 p_i；对每个粒子，将其适应值与整个粒子群所经历过的最优位置 p_g 的适应值进行比较，如果更好，则将该粒子作为新的 p_g ；

f)如果没有满足最终条件（适应值小于预设阈值或者超过最大迭代次数），则返回步骤 b)，否则，退出算法，输出寻优过程曲线和最优解。

使用改进的粒子群算法对于自抗扰控制器的参数进行优化，其过程如图 5 所示。

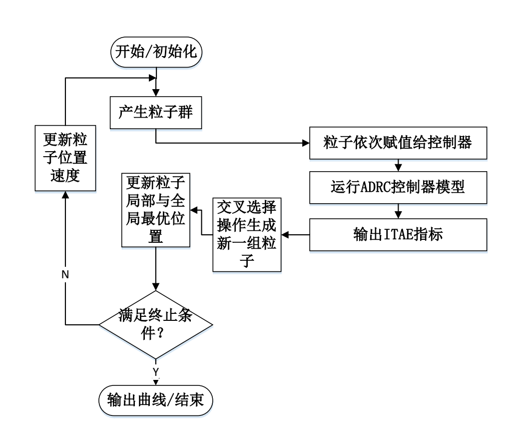


图 5　PSO 算法流程图

关于惯性权重 ω，较高的 ω 会使 PSO 拥有更好的全局搜索能力，而低 ω 值则对于局部搜索更加有利[16]。为了使得粒子群算法在计算初期具有更好的全局搜索能力，而在计算后期具有更好的局部寻优能力，现将 ω 设置为如下动态形式：

$$\omega(k)=\omega_{start}-(\omega_{start}-\omega_{end})\left(\frac{k}{T_{max}}\right)^2$$

其中，$\omega(k)$ 为第 k 次迭代时的惯性权重，ω_{start} 为惯性权重初始值，通常设定为 0.9，ω_{end} 为惯性权重终值，通常设定为 0.4，T_{max} 为最大迭代次数。

4　实验分析

本文首先建立四旋翼 ADRC 控制器模型并与传统串级 PID 控制器进行比较，ADRC 控制器仿真模块如图 6 所示：

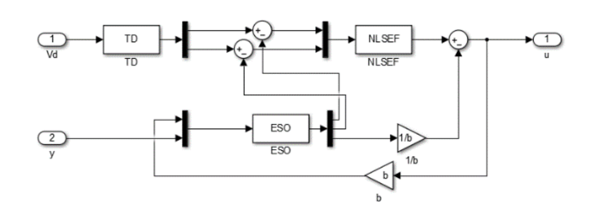


图 6　Matlab/Simulink 中 ADRC 仿真模块

仿真实验以滚转通道为例，旋翼自身参数如表 1 所示。

表 1　飞行器仿真参数

参数	数值
m/kg	1.5
l/m	0.25
$g/(m/s^2)$	9.81
$I_x/(kg\cdot m^2)$	0.033
$I_y/(kg\cdot m^2)$	0.033
$I_z/(kg\cdot m^2)$	0.061
$k_1/(N\cdot s^2)$	3.1×10^{-7}
$k_2/(N\cdot s^2)$	1.12×10^{-7}

选取控制周期为 0.01s，ADRC 参数如表 2 所示。

表 2　ADRC 仿真参数

模块	参数	数值
TD	r	2
	h	0.05
ESO	β_1	100
	β_2	450
	β_3	2900
	α_1	0.5
	α_2	0.25
	δ	0.1
NLSEF	k_1	80
	k_2	40
	α_1	0.75
	α_2	1.25
	δ_0	0.1
	b	5

设定期望滚转角度为 15 度，分别仿真 PID 和 ADRC 两种控制器得到的输出曲线如图 7 所示，控制性能指标如表 3 所示，从图中可以看出自抗扰算法超调量小于串级 PID 控制器且调节时间更短。

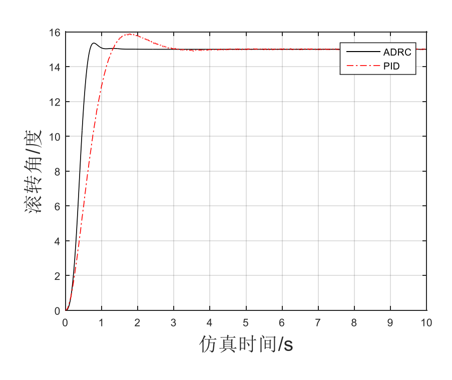


图 7　ADRC/PID 控制效果对比

表 3　滚转角控制性能指标对比

控制器	超调/%	调节时间/s
串级 PID	5.89	3.09
自抗扰控制	2.31	1.64

为了测试 ADRC 控制器的抗干扰能力，对两种控制器加入高斯白噪声，其控制效果如图 8 所示。

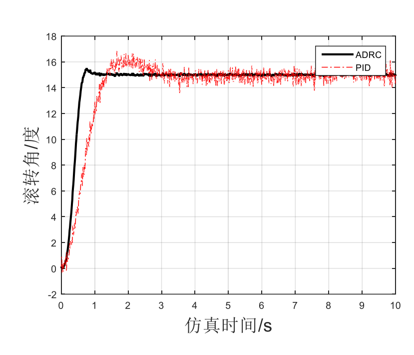


图 8　加入高斯噪声后 ADRC/PID 控制效果对比

飞行器稳定状态下，在 5s 时刻加入幅值为 1、脉宽为 1s 的干扰输入信号来测试两种控制器的抗干扰能力，控制效果如图 9 所示。

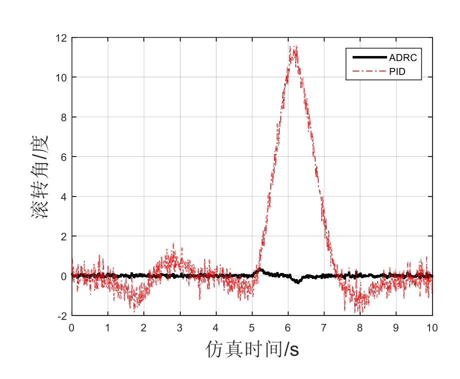


图 9　加入突变噪声后 ADRC/PID 控制效果对比

实验表明，ADRC 具有优秀的噪声抑制和抗干扰能力。

之后，加入改进粒子群对 ADRC 参数进行优化，对粒子群算法的各项参数进行设定，具体数值如表 4 所示。

表 4　PSO 参数设定

参数	数值 $[k_1,k_2,b,\beta_1,\beta_2,\beta_3]$
c_1	2
c_2	2
粒子维度	6
粒子群规模	50
最大迭代次数	50
粒子速度上限	[1,1,0.25,1,2,5]
粒子速度下限	[-1,-1,-0.25,-1,-2,-5]
粒子寻优上限	[400,400,30,200,1000,4000]
粒子寻优下限	$\left[1,1,\frac{1}{30},60,200,1500\right]$

以四旋翼滚转角通道为例，其他参数设定保持不变，只进行 $k_1,k_2,b,\beta_1,\beta_2,\beta_3$ 的优化，以 15 度作为输入，优化后与之前参数的各性能指标如表 5 所示。

表 5　滚转角优化前后参数与控制性能对比

	优化前	优化后
β_1	100	94.13
β_2	450	437.89
β_3	2900	2930.31
k_1	80	253.48
k_2	40	212.15
b	5	9.87
超调/%	2.31	0.32
调节时间/s	1.64	1.25
ITAE 性能	0.0145	0.0016

各个参数的变化曲线与优化之后的仿真结果如图 10~13 所示。

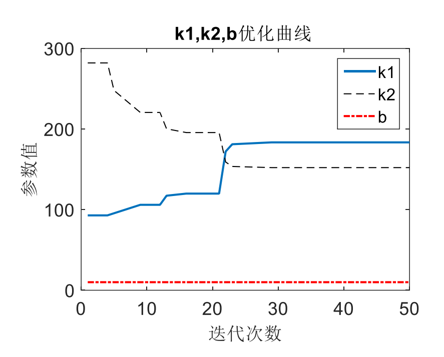


图 10　k_1,k_2,b 优化曲线

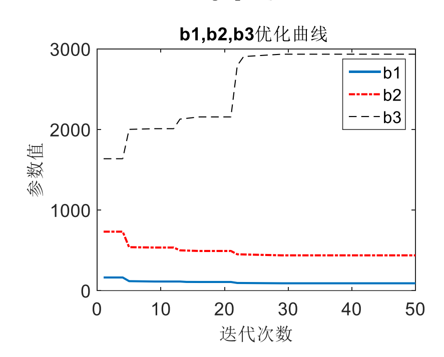


图 11　β_1,β_2,β_3 优化曲线

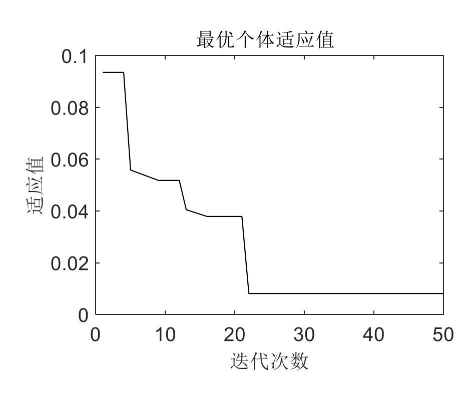


图 12　ITAE 适应值曲线

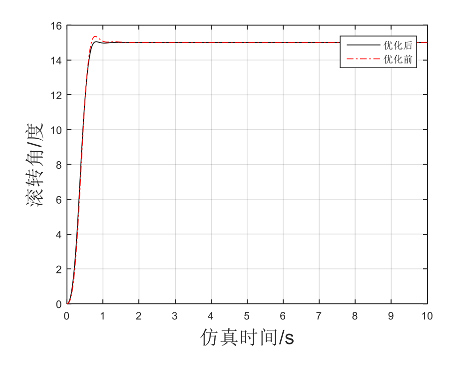


图 13　优化前后的滚转角控制曲线

俯仰角通道的特性与滚转角相近，可以使用相同的参数，不再另行优化。航偏角和高度分别输入 60 和 5，优化前后的参数分别如表 6、7 所示，对比曲线分别如图 14、15 所示。

表 6　航偏角优化前后参数与控制性能对比

	优化前	优化后
β_1	100	112.45
β_2	450	609.71
β_3	3000	3532.56
k_1	100	288.14
k_2	80	256.2
b	15	23.41
超调/%	3.23	0.79
调节时间/s	2.75	2.11

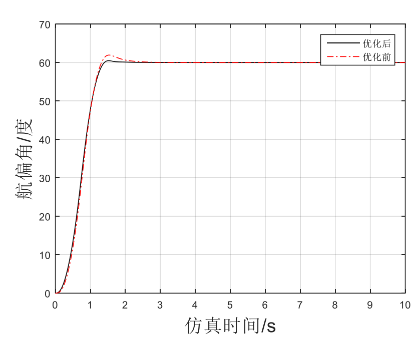


图 14　优化前后的航偏角控制曲线

表 7　高度优化前后参数与控制性能对比

	优化前	优化后
β_1	100	127.14
β_2	450	771.28
β_3	3000	4000
k_1	120	322.4
k_2	100	335.21
b	20	25.63
超调/%	3.68	0.8
调节时间/s	4.36	3.42

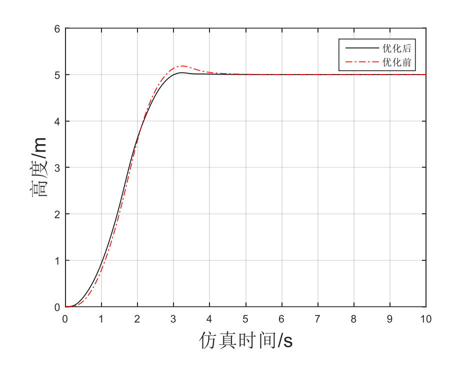


图 15　优化前后的高度控制曲线

通过仿真实验可以看出，经过改进 PSO 算法优化后的 ADRC 控制器超调更且调节时间更短，同时无须人工调参。和滚转角通道类似，俯仰角通道、航偏角通道和高度通道的控制效果经过改进 PSO 算法优化后也具有很好的效果。因此使用改进粒子群算法对 ADRC 控制器进行参数优化明是一种行之有效的方法。

5　结束语

本文首先设计了基于自抗扰控制技术的四旋翼飞行器姿态与高度控制器，为了得到最优控制效果，并针对自抗扰控制器参数较多，人工整定耗时长且难以达到最优的问题，提出了基于粒子群算法的自抗扰控制器参数优化方法，同时为了解决粒子群算法易陷入局部最优的问题，加入遗传算法中的交叉操作，对适应值不好的粒子进行交叉保优，以提高粒子的多样性，加快寻优速度。使用 Simulink 对整个优化过程进行了仿真，结果表明，使用改进的 PSO 算法，能够有效解决自抗扰控制器人工参数整定困难的问题，经过改进的 PSO 算法优化后的自抗扰控制器具有更好的动态特性和稳态特性，且算法简单，调参时间短，易于实现，具有较大的实际应用价值。

参考文献：

[1] Gonzalez-Vazquez S, Moreno-Valenzuela J. A new nonlinear PI//PID controller for quadrotor posture regulation [C]// Proc of Electronics, Robotics and Automotive Mechanics Conference. 2010: 642-647.

[2] Alothman Y, Jasim W, Gu Dongbing. Quad-rotor Lifting-transporting Cable-suspended Payloads Control [C]// Proc of the 21st International Conference on Automation and Computing. 2015: 1-6.

[3] Yang Ning, Wu Liaoni, Lin Qi. Adaptive flight control law based on neural networks and dynamic inversion for unmanned aerial vehicle [C]// Proc of International Conference on Automatic Control and Artificial Intelligence. 2012: 355-358.

[4] Raffo G V, Ortega M G, Rubio F R. Robust nonlinear control for path tracking of a quad-rotor helicopter [J]. Asian Journal of Control, 2015, 17 (1): 142-156.

[5] Tan Lining, Lu Libin, Jin Guodong. Attitude stabilization control of a

quadrotor helicopter using integral Backstepping [C]// Proc of International Conference on Automatic Control and Artificial Intelligence. 2012: 573-577.

[6] De monte P, Lohmann B. Trajectory tracking control for a quadrotor helicopter based on backstepping using a decoupling quaternion parametrization [C]// Proc of the 21st Mediterranean Conference on Control & Automation. 2013: 507-512.

[7] 刘一莎, 杨晟萱, 王伟. 四旋翼飞行器的自抗扰飞行控制方法 [J]. 控制理论与应用, 2015, 32 (10): 2-5.

[8] 杨立本, 章卫国, 黄得刚. 基于ADRC姿态解耦的四旋翼飞行器鲁棒轨迹跟踪 [J]. 北京航空航天大学学报. 2015, 41 (6): 1026-1033.

[9] 叶孝璐, 俞立, 张文安等. 基于串级 ADRC 的四旋翼飞行器悬停控制 [J]. 中南大学学报: 自然科学版, 2017, 48 (8): 2079-2087.

[10] 韩京清. 从 PID 技术到"自抗扰控制"技术 [J]. 控制工程, 2002, 9 (3): 13-17.

[11] 张立明. 自抗扰控制技术在 AUV 航向控制中的应用 [D]. 哈尔滨: 哈尔滨工程大学, 2009.

[12] Kennedy J, Eberhart R. Particle swarm optimization [C]// Proc of IEEE International Conference on Neural Networks. 1995: 1942-1948.

[13] Eberhart R, Kennedy J. A New Optimizer Using Particle Swarm Theory [C]// Proc of the 6th International Symposium on Micro Machine and Human Science. 1995: 39-43.

[14] 张万绪, 张向兰, 李莹. 基于改进粒子群算法的智能机器人路径规划 [J]. 计算机应用, 2014, 34 (02): 510-513.

[15] 田佳, 胡威, 李琳, 柯鹏, 张凯. 基于粒子群优化算法的多核处理器任务调度研究 [J]. 计算机应用研究, 2017, 34 (12): 3698-3700.

[16] Shi Yuhui, Eberhart R. A Modified Particle Swarm Optimizer [C]// Proc of IEEE International Conference on Evolutionary Computation. 1998: 69-73.